

REMARKS

The application includes claims 1-22 prior to entering this amendment.

The examiner rejects claims 1-22 under 35 U.S.C. § 102(e) as being anticipated by Dworkin et al. (U.S. Pub 2003/0058893).

The applicant amends claims 1, 6, and 10. The application remains with claims 1-22 after this amendment.

The applicants add no new matter and request reconsideration.

Claim Rejections - 35 U.S.C. § 102

Applicant's synchronization circuits operate in a way fundamentally different than those of Dworkin et al. U.S. Pub. 2003/0058893, the art now at issue, or, for that matter, of Dworkin Int'l Pub. WO 01/17167, which art was previously cited by applicant.¹ In applicant's circuits, the synchronization events are activated by what may be termed "global" synchronization pulses. In contrast, in Dworkin '893 the synchronization event is self-triggered or internally activated by each synchronization circuit (master or slave) acting independently of one other. In Dworkin '167, the synchronization event is activated by a "control" pulse generated by the master counter and conveyed over a bus universally designed to carry control and data signals exchanged between the CMTS (master and slave) devices. It is helpful to keep these basic differences in mind as one considers the remarks below, which focus on the limitations of particular claims, how these limitations distinguish over the art, and the implications such distinctions have on operational performance.

Independent claim 1

Claim 1 calls for a synchronization circuit that includes a local timestamp counter and a processing circuit designed to synchronize this counter with a predicted master timestamp value. In keeping with the above mentioned principle of a "global" synchronization signal, amended claim 1 calls for the processing circuit to receive externally generated synchronization pulses and

¹ Applicant also cited Int'l Pub. WO 03/028374, but this art is cumulative with Dworkin et al. 2003/0058893 cited by the examiner.

to perform its synchronizing function “upon receipt” of a “future one” (associated with a predicted master timestamp) “of the externally generated synchronization pulses.” Support for this language is provided by Figs. 1 and 2 of applicant’s original drawings, which show various synchronization circuits 18A, 18B, and so on, each receiving, from central time source 12, externally generated sync pulses 14. By using a selected (future) one of these pulses to activate the synchronization operation of each sync circuit, the claimed design ensures that the corresponding counters are effectively synchronized concurrently.²

The limitations just emphasized serve to distinguish claim 1 over Dworkin ‘893. In the synchronization circuits of Dworkin ‘893, the pulse activating the synchronization event is internally generated independently by each circuit (and is not, as claimed, “received” as an externally generated pulse). This will now be confirmed by reference to Fig. 3 of Dworkin ‘893, which depicts the circuitry of an individual synchronization circuit 223 (par. 0053).

Referring to Fig. 3, each synchronization circuit 223 of Dworkin ‘893 includes a counter 315 that begins generating timestamp values (e.g., begins generating a multibit word representing a running timecount) when triggered by an external “calibration” pulse (pars. 0076-0077). Dworkin notes that each counter will likely finish calibrating at a different time so that the “base” timestamp (as verified at some “base” reference time) will likely vary from counter to counter (par. 0076 and second-to-last sentence of par. 0078). In order to synchronize the counter of each circuit so that each provides effectively the same “current” time, each circuit includes a compare register 305 and compare circuit 306. These elements, acting together, perform what might fairly be called an internal countdown operation; that is, starting with the unique “base” value of their circuit’s counter, elements 305 and 306 effectively count down (or, to be more precise, up) “Z” clock cycles and then issue an internal synchronization pulse (“Intpulse” in Fig. 3) to reset this counter to a prearranged “new” value. More specifically, each compare register 305 is loaded with a target value equaling the base value of the corresponding device counter 315 plus “Z” clock cycles (par. 0084), so that when this counter 315 subsequently reaches this “base

² As will be understood by those of ordinary skill in light of applicant’s teachings, as a practical matter, slight differences in sync line and gate delays will result in slight differences in timing of the synchronization event between different timestamp counters 44A, 44B, etc.; however, these path differences, being physically based, are substantially invariant and can therefore be fully offset or corrected if each corresponding processing circuit 40A, 40B, etc. makes an appropriate “fixed” adjustment to the received “predicted master timestamp value” before storing this value in the corresponding holding register 42A, 42B, etc. See, generally, applicant’s original disclosure, page 5, lines 14-21. In other words, in the claims, use of the word “with” in the phrase “synchronize the...counter with the predicted master timestamp value,” is intended to be interpreted as meaning “with use of” or “based on.”

plus Z” target value (presumably at the same time as the other counters, that is, after precisely Z clock cycles; see the last two sentences of par. 0083), the corresponding compare circuit 306, detecting the congruency, issues the internal pulse (intpulse; pars. 0079, 0089, and the first sentence of par. 0090) that resets this counter 315 to the prearranged “new” value. This “new” value, which was previously stored in each circuit in load register 310 for ready loading into counter 315, constitutes, in a preferred embodiment, the base timecount of the master plus “Z” clock cycles, that is, it is the “predicted” timecount of the master circuit after “Z” clock cycles (see pars. 0085, 0088, and the various formulations of the future time stamp value or “FTSV” discussed below).

From the foregoing operational summary of the sync circuits 223 of Dworkin ‘893, it will be apparent that this reference fails to constitute a synchronization circuit of the type recited in claim 1. For example, although Dworkin indeed includes a “processing circuit” 222 for executing “software functions” at the headend MAC 218 (par. 0048), there is no mention that one of these functions includes receiving “synchronization pulses” for a synchronization circuit 223, as claimed. In fact, as just described, each synchronization circuit 223 uses its own internally generated synchronization pulse (intpulse) to reset its own counter and provide synchronization with the other counters and not, as claimed, a “received” or externally generated synchronization pulse.

Dworkin ‘893 does speak of a “synchronization bus” 112 (see Fig. 1). The purpose of this bus is to provide a suitable path for “the transmission of electronic pulses and data between calibration pulse generator 103, CMTS 104A, CMTS 104B, and CMTS 104C.” While the electronic pulses directed to each synchronization circuit 223 may, collectively, lead up to the internal countdown that does eventually reset and synchronize each counter, not one of these pulses, acting alone, suffices to cause “the processing circuit to ...synchronize the local ... counter ... upon receipt,” as claimed. For example, the calibration pulse produced by generator 103 (Fig. 1), as noted above, only serves to initialize each counter 315 and start it running at some variable “base” value (par. 0076). Further pulses (including for clocking and multiplexing) are needed to load the “verify” register so that each base value can be read (par. 0078), to load the “compare” register 305 with the base value plus Z (par. 0084), and to load the prearranged “new” value into the load register 310 (par. 0085 and 0088). In Dworkin ‘893, the only pulse that is sufficient, by itself, to “synchronize the local ...counter ... upon receipt,” as claimed, is the

internal pulse “Intpulse” (Fig. 7), and this pulse is not an external pulse, as also claimed (see above remarks and pars. 0079 and 0089-0090). In short, while one might reasonably refer to certain external pulses in Dworkin as “synchronization” pulses (such as the calibration pulse), such synchronization pulses are not of a type fully meeting the criteria specified in claim 1.

The basic difficulty with Dworkin’s approach in ‘893 is that it attempts to synchronize the timecounts of the master and slave circuits while assuming that the counter in each circuit generating these timecounts is impervious to operational glitches or error. If this were truly the case, it would only be necessary to synchronize the master and slave circuits once, say during initial equipment setup, and then rely on their error-free operation to keep them in perfect synchronization for all times thereafter. Indeed, Dworkin makes explicit reference to this notion where it states “[b]ecause each of the CMTS devices 104 are run from the same oscillator” (i.e., clock) “it is presumed that each will remain in sync with the other for relatively long periods of time” (par. 0092). In other words, despite teaching a particular approach to synchronizing the master and slave circuits, Dworkin simultaneously discounts the need for such synchronization, at least where a universal clock is used.

As an example of the type of problems that may arise when using the Dworkin ‘893 system, consider a situation in which some glitch, say spurious clocking signals caused by excessive crosstalk or a power spike, causes one of Dworkin’s slave counters, shortly after it is calibrated and set to its initial “base” value, to advance two clock cycles ahead of where it should be. This would cause the slave counter to reach the “base plus Z” value two clock cycles too early and to prematurely load, again by two clock cycles, the predicted master timecount before the same master timecount is loaded into the master counter. Accordingly, the respective counters of the master and slave devices would not synchronize at the same value at the same moment, so that even from the first, their respective timecounts would diverge until at least the next synchronization event. Hence these devices would not be able to properly support each other in servicing, for example, a particular cable modem.

In contrast to Dworkin’s ‘893 synchronization circuit, the synchronization circuit of claim 1 does not rely on the workings of an internal process to initiate the synchronization event but rather relies on “received” synchronization pulses. In particular, amended claim 1 clarifies that these “received” synchronization pulses are “externally generated” and that the local counter is synchronized with the predicted master timestamp value “upon receipt of the future one of the

synchronization pulses” with which the predicted master timestamp value was previously associated.

Faced, then, with a fault situation of the type described above, in which a counter in a slave synchronization circuit (say 44B in applicant’s Fig. 2) erroneously skips ahead in its timecount, the synchronization circuit recited in claim 1 remains immune to premature triggering of the synchronization event. In particular, whether or not the counter has improperly advanced two clock cycles ahead prior to synchronization will make little difference once synchronization occurs because, in the claimed circuit, synchronization will still be initiated “upon receipt” of the future one of the externally generated synchronization pulses. This synchronization pulse, being externally generated, can likewise be used to initiate the synchronization event for the master synchronization circuit (indeed, it may originate with the master circuit; see applicant’s specification, page 3, lines 24-25 and page 5, line 11) so as to ensure that all the circuits, master and slave, effectively synchronize concurrently.³ Moreover, provided that these externally generated sync pulses are also centrally generated, as shown in Fig. 1, then even if the skipping counter is located in the circuit producing the sync pulses, that is, even if the sync pulses accelerate or decelerate relative to the system clock pulses, all that will happen is that the synchronization event will be advanced or delayed an equal period for all the synchronization circuits and, as a result, premature triggering by any one of the circuits is avoided.

Before turning to the remaining claims, applicant would point out that claim 1 contains further limitations patentably distinguishing over Dworkin ‘893. Dworkin does not suggest that the synchronizing circuit receive, at a processing circuit, the predicted master timestamp value asynchronously in Internet Protocol (IP) packets received over an IP connection. It is true that in order to provide cable modem (CM) users with high speed connection to the Internet, cable modem termination systems (CMTS’s) traditionally relay user data to an upstream Internet access or exchange point (this is analogous to a telephone system relaying DSL data to an upstream exchange point) . The question, however, is not whether Dworkin’s system facilitates user connection to the Internet; clearly it does since it allows Ethernet packets “recovered” by the MAC 218 to be transferred to an Ethernet interface 224 for delivery to a packet-switched network via a router (par. 0049). The relevant question is whether this network interface serves a dual-purpose, that is, whether it is being used not only in the conventional manner to facilitate

³ See footnote 2 supra.
AMENDMENT

high speed data connection by the user but also whether it can be used, as claimed, to enable a processing circuit (e.g., in the slave sync circuit) to receive the predicted master timestamp value in asynchronous manner in Internet Protocol (IP) packets. Just because Dworkin shows an IP network interface provided in a CMTS system for use in a conventional manner doesn't necessarily mean it supports use of this same interface in a nonconventional manner.

In fact, Dworkin '893 merely states that the predicted master timestamp value, which Dworkin calls the "TGCCCount(new)" value, "is loaded into its TGCLoad register 310" (par. 0088, first sentence) without explicitly mentioning where this predicted master timestamp value comes from, though it does offer various formulations suggesting this value is calculated by a processor using operands provided, at least in part, by the master circuit. For example, it notes that at some future point, the TGCCounter for the master (that is, its running value) will equal the master's future time stamp value (or "FTSV") and that it is this "future" value to which the slave CMTS devices should be set (par. 0085). In other words, the slave CMTS devices should be set to a value equaling the master's TGCCCount(base) + Z (see par. 0082) with Z identifying the number of clock cycles needed to reach this future point. The same result obtains by taking the difference between the TGCCCount(base) of the master and the TGCCCount(base) of the slave (each TGCCCount(base) value being verified at the verify register 320 of the corresponding circuit) and then adding back the TGCCCount(base) of the slave, adding Z, and adding some "latency" factor to compensate for the time needed to deliver the master value to the slave (par. 0087). This last formulation, if anything, suggests that Dworkin envisions a fixed path (with fixed latency) between the master and slave circuits, such as a dedicated line or perhaps the "synchronization bus" 112 that Dworkin '893 makes available for "data" transmissions between CMTS devices (par. 0041, last three sentences). There is no suggestion, in Dworkin, of dividing the predicted master timestamp value into "packets," as recited in claim 1, and sending these packets over an "IP connection" for asynchronous receipt by the slave synchronization circuit.

Thus far, the remarks in this section have been directed toward Dworkin et al. U.S. Pub. 2003/0058893. However, claim 1 also distinguishes over the other art of record, Dworkin et al. Int'l Pub. WO 01/17167. Like Dworkin '893, Dworkin '167 fails to show or suggest that a processing circuit in the synchronization circuit receives the predicted master timestamp value asynchronously in Internet Protocol (IP) packets received over an IP connection.

In Dworkin '167, a synchronization circuit 300 (Fig. 7) is included in each CMTS device (master or slave, page 10, lines 2-3). A synchronization bus 30 connects each device (master and slave) with a CPU 311 and the devices with each other (Fig. 3; page 5, lines 1-2). During operation, the CPU "polls" the master device for the current value of the master device's counter 202 (Fig. 7; page 10, lines 29-30) and, by adding a predetermined number of cycles to that value, generates a future time stamp value (page 10, lines 35-36) to be used for synchronizing the counters of each device (both master 22 and slave 24; page 5, lines 14-17). The CPU "broadcasts" this future timestamp value to each device (master and slave) over bus 30 (page 2, lines 18-19; page 5, lines 4-6 and 17-19; page 10, lines 35-37) as a 32-bit word called "TSLoadVal" (page 5, line 19; page 10, line 37; the suffix "[31:0]" appended to "TSLoadVal" in Fig. 7 indicates bit size) where it is received by a loading register 304 and is thus available for ready loading into each device's counter 202 when these devices gets the "synchronization" command.

The synchronization command is a pulse, "LdTsExt," generated by the master device when the continually incrementing "current" value of its counter 202 subsequently matches a target value held in the master's comparator register 302 (page 10, lines 9-15). Nominally, this target value is the FTSV (or TSLoadVal) broadcast earlier by the CPU, that is, the master sends out the command for each counter to synchronize to the FTSV value when its own counter reaches the FTSV value. In practice, however, there are delays between the exact moment the match occurs (when the master's counter reaches the activate value) and the exact moment the resulting synchronization command resets each counter to the FTSV value. Referring to Fig. 7, these include gate delays through elements 310 and 312 of the master device, minor path delay as the LdTsExt command travels between devices along bus 30, and further gate delays through elements 314 and 316 in the device receiving the synchronization command. Accordingly, the target value that the CPU loads into the master's comparator register 302 is "deliberately selected to be one or two cycles behind" the FTSV value that the CPU loads into the loading register 304 of each device (page 11, lines 4-11). Timed correctly, the master's own counter (and the other slave counters) will receive the instruction to reset to the FTSV value at the same moment the current value of the master counter would otherwise be changing, in accordance with normal orderly progression, to the FTSV value (so as to prevent the master counter value from jumping

forward or backward at the precise moment of synchronization, which discontinuity might disturb any time-sensitive communications then occurring between the master and user devices).

In the Dworkin '167 setup just described, the universal "synchronization" bus (item 30 in Fig. 3 and "UbusSlave" in Fig. 7) conveniently handles all CPU-to-device and device-to-device communications (page 2, lines 16-21; page 5, lines 1-6), including "data" transmissions (e.g., broadcast of the FTSV by the CPU; see page 5, lines 14-21 and page 10, lines 35-38) and "control" transmissions (e.g., transmission of the synchronization pulse by the master device; page 2, lines 19-21 and page 5, line 39 to page 6, line 1). However, it will be noted that the control pulse, LdTsExt, generated by the master device must propagate between circuits with a substantially fixed delay in order to correspond with the delay represented by the respective values loaded into comparator 302 (the target value) and the load register 304 (the future master timestamp value). This, in turn, locks Dworkin into using a synchronization bus having substantially fixed delay characteristics or, to put it another way, precludes Dworkin from using a synchronization bus possessing delay characteristics that are not well defined. Accordingly, Dworkin '167 cannot simply use an IP network for its universal synchronization bus (as opposed, for example, to a fixed path such as a "back-plane bus, four-wire interface, coaxial cable, or even a wireless [fixed endpoint-to-endpoint] path" see page 5, lines 7-11), because, in an IP network, the IP packets travel along dynamically switched paths and thus reach their destination at imprecisely defined times and possibly even out-of-order (which is why a transport layer protocol, such as TCP, is traditionally used to ensure correct reassembly of IP packets). Thus, comparing the Dworkin '167 circuit to that specified in claim 1, Dworkin does not show or suggest a processing circuit that "receives the predicted master timestamp value" (e.g., the FTSV) "in Internet Protocol (IP) packets received over an IP connection."

In accordance with the above considerations, it is respectively submitted that claim 1 patentably defines over Dworkin et al. U.S. Pub. 2003/0058893 and, furthermore, patentably defines over Dworkin et al. Int'l Pub. WO 01/17167.

Dependent claims 2-9 and 21-22

Dependent claims 2-9 and 21-22 incorporate each limitation of independent claim 1 and so patentably define over the art of record for the same reasons given above. However,

dependent claims 3, 6, and 7, even independently of their base claim 1, patentably define over the art of record.

Dependent claim 3 specifies that the Internet Protocol (IP) packets containing the master timestamp value use a multicast address. In the sense used in applicant's disclosure, what this signifies is that a plurality of informational packets are received, e.g., by respective slave circuits, even though only "one" informational packet is sent out (see applicant's disclosure, page 4, lines 11-12). In other words, the sent packet is replicated somewhere within the network itself, for example, at the "multicast" address, where it is copied to a targeted group of links leading to the slave devices forming the group. One advantage of this approach, as noted in applicant's last amendment, is that the network bandwidth required to carry the master timestamp to each slave is reduced (e.g., over a system where the packet received by each slave device traverses the entire network from end-to-end). Though the synchronization bus shown in Dworkin et al. '893 or '167, as item 112 (Fig. 1) or 30 (Fig. 3), respectively, can "broadcast" a master timestamp value to many slave devices at once, this bus, for reasons noted above, provides a fixed path (with fixed latency or delay) and is neither a network that provides dynamic switching of IP packets nor, as claim 3 further requires, includes an intermediate node capable of supporting a "multicast" function (e.g., an "active" node capable of replicating the master timestamp information). Thus dependent claim 3 patentably defines over the art of record even independently of claim 1.

Dependent claim 6, as amended, requires that the synchronization pulses cycle at a rate of somewhere between 8 kHz and 1 Hz. In Dworkin '893, the synchronization pulse is internally generated by each device using an internal "countdown" process (e.g., see the first three sentences of par. 0091 and the above discussion of claim 1) and is not generated by the calibration generator (which only provides a "calibration" pulse whereby the respective counters are started up and "initialized" at some "base" count normally differing from device-to-device; see first two sentences of par. 0087). Even assuming, for argument sake, that it is a synchronization pulse being generated by Dworkin's calibration pulse generator 103, paragraph 0039 cited in the Action makes clear this generator generates "a single pulse of specific width, at a specific time" (e.g., not a train of pulses in which individual pulses recur at a regular rate; compare applicant's recurring sync pulses shown in Fig. 3 of applicant's drawings). This "single" pulse is consistent with Dworkin's notion that, once synchronized, the devices will

remain in sync with each other “for relatively long periods of time” with a universal clock (par. 0092; a one-shot approach and similar language is found in Dworkin ‘167, page 8, lines 15-17). Even assuming, once again for argument sake, that Dworkin’s generator 103 produces individual pulses that repeat at a regular rate, paragraph 0039 being cited gives the pulse time as 90 nsec. Taking the inverse of this pulse time works out to a cycle rate of 11.1 MHz (not between 8 kHz and 1 Hz, as claimed). Hence dependent claim 6 patentably defines over the art of record even independently of claim 1.

Dependent claim 7 calls for a synchronization circuit in which an error condition is identified, by the processing circuit, according to a number of times the local timestamp counter is synchronized with received timestamp values. It should be noted, here, that a threshold “number” (item 102 in applicant’s Fig. 5) is used because devices may go out-of-sync not due to inherent defect but rather due to occasional spurious signals. This claimed approach of determining the actual (versus nominal) counter resets (in which the “local” counter value is actually replaced by the “received” value) and permitting these to reach some threshold “number” before identifying an “error condition” is simply not discussed in the Dworkin ‘893 (or ‘167) reference. At best, Dworkin ‘893 notes, in passing as it were, that if the master device fails, one or more of the slave devices can assume the load (see second-to-last sentence of par. 0091; Dworkin ‘167 has similar language on page 6, lines 6-9). However, Dworkin fails to describe any particular criteria for identifying device failure (in either the master or slave devices). Dworkin does describe “periodically” checking a verify register 320 of each device (which holds the value of the device’s counter at its last reset) and comparing these values to verify they are all identical, but if these values are not identical, Dworkin recommends simply that the synchronization method be “repeated” (see par. 0093) from which one may reasonably conclude that some other criteria or process is involved in identifying device failure in Dworkin. In Dworkin, nothing is said about tracking the number of resets in which the local counter value is actually changed to (or synchronized with) a received value (the predicted master timestamp value) and identifying an error condition if this exceeds a threshold “number of times.” As applicant additionally teaches, it is further desirable to track this number relative to some limited number of correct sync events using, for example, a decrementing error value (see applicant’s Fig. 5). For these reasons, then, dependent claim 7 patentably defines over the art of record even independently of claim 1.

Independent claim 10

Claim 10 is directed to the master (versus slave) synchronization circuit. Applicant has amended claim 10 to better clarify how applicant's master circuit differs from that taught in Dworkin '893 (or '167). Recalling, from the overview introducing these remarks, the significance that "global" synchronization signals have to applicant's invention, claim 10 has been amended to define, with particularity, the role played by these synchronization pulses in respect to the master synchronization circuit.

Claim 10, in particular, now separately identifies the synchronization pulses and clocking signal (see applicant's Fig. 1, items 14 and 15, respectively). Describing the role of the clocking signal, claim 10 now calls for a master counter to generate respective master timestamp values varying with cycling of the clocking signal (see applicant's Fig 3, item 15). Describing the role of the synchronization pulses, claim 10 calls for a processing circuit to determine, for a given count of consecutive ones of synchronization pulses cycling less often than the clocking signal, the corresponding difference between the master timestamp values. (In applicant's disclosure, the given count is referred to as the "wait count;" see applicant's disclosure, page 7, lines 25-27; consecutive ones of sync pulses are referred to as "adjacent" ones, see page 8, lines 7-9; and the "corresponding difference" phrase is a shorthand way of talking about the difference between the number of master timestamp values, which vary with the relatively fast clocking signal, and the consecutive sync pulses, which vary "less often," hence, in the example in applicant's disclosure on page 6, lines 16-23, the "given count" is one consecutive sync pulse following the starting sync pulse, for which the "corresponding difference" is five clock or timestamp value cycles; the term "difference" also reuses the same distinguishing term found in the form of claim 10 immediately preceding the current amendment). Claim 10 goes on to detail how the "future master timestamp value" is calculated (i.e., by "adding the corresponding difference to an initial one of the timestamp values corresponding to an initial one of the synchronization pulses;" hence, in the example in applicant's disclosure on page 6, lines 16-23, the corresponding difference 5 is added to initial timestamp value 35 corresponding to initial sync pulse 52 to calculate a future master timestamp value of 40). The claim concludes by specifying that the future master timestamp value is forwarded to a slave synchronization circuit "over a wide area network" (applicant's disclosure, page 4, lines 1-4) with the result that "a slave counter in the

slave synchronization circuit” (e.g., applicant’s Fig. 2, counter 44B in slave circuit 18B) is desirably synchronized “with the future master timestamp value” at “a future synchronization pulse generated independently of operations by the master counter and slave counter” and “corresponding to the given count of consecutive ones of synchronization pulses following the initial one of the synchronization pulses” (thus, in the example given in applicant’s disclosure on page 6, lines 16-23 and depicted in applicant’s Fig. 3, a slave counter is synchronized with the future timestamp value 40 at a future synchronization pulse 54 corresponding to one consecutive sync pulse following the initial sync pulse 52.)

From the description just provided, a number of differences will be recognized between the master synchronization circuit defined in claim 10 and that taught in Dworkin ‘893 (and ‘167). For example, claim 10 calls for a clocking signal and synchronization pulses “cycling less often” than the clocking signal. Dworkin only describes the clocking signal without specifically identifying synchronization pulses that cycle at a lower rate (note that Dworkin’s clocking signal, being of higher frequency, is more susceptible to crosstalk and other ill-controlled effects caused by spurious radiation). Claim 10 calls for the processing circuit to determine, based on a given count of consecutive ones of the synchronization pulses, the corresponding difference between the master timestamp values. No such difference is calculated in Dworkin ‘893 (and ‘167). Instead, a processor in Dworkin establishes, based on a given count of clocking (not sync) signals and the starting value of a counter in at least one of the synchronization circuits, at least one target value for synching (note that this target value itself depends on the operation of at least one counter). In Dworkin ‘893, a customized target value (FTSV) is set for each different synchronization circuit (master or slave) based on adding Z clocking cycles and the starting (base) value of the respective counter in the corresponding circuit (par. 0084). In Dworkin ‘167, only one target value (TSLoadVal minus “one or two” clock cycles; see page 11, lines 7-9) is set based on adding a predetermined number of clocking cycles and the starting (current) value of the master counter (page 10, lines 35-39).

Claim 10 further calls for the master circuit to forward the calculated future master timestamp value to a slave synchronization circuit over a wide area network. In Dworkin, both the future master timestamp value and the above-mentioned target value are forwarded over a fixed-delay data path or bus, to at least one of the synchronization circuits (note that Dworkin needs to also forward target values because, as mentioned, there are no synchronization pulses

independently available). In Dworkin '893, the future master timestamp value (TGCCCount(new)) and a customized target value (FTSV) are sent to each sync circuit presumably over bus 112 designed to handle data exchanges between sync circuits (par. 0041, second-to-last sentence). As noted in the comments for claim 1 above, the future master timestamp value (TGCCCount(new)) is calculated based on a fixed "latency" between the master and slave circuits, that is, based on a fixed delay in the data path or bus (see par. 0085). In Dworkin '893, each circuit uses the customized target value (FTSV) to internally generate its own sync pulse (intpulse, Fig. 7) and to trigger the synchronization event. In Dworkin '167, though the future master timestamp value (TSLoadVal) is sent to each circuit, only the master circuit receives the target value (TSLoadVal minus one or two cycles; see page 11, lines 7-9), and this master circuit, based on the target value, generates the synchronization pulse (LdTsExt) which is then routed to each synchronization circuit (the master and each slave) to trigger the synchronization event (page 10, line 38 to page 11, line 3). In Dworkin '167, a universal "synchronization" bus (item 30 in Fig. 3 and "UbusSlave" in Fig. 7) forwards the data values (both the target value loaded into master register 302 and the future master timestamp value loaded into each register 304) from the CPU to the circuits and also forwards the sync (or "control") pulse from the master circuit to each sync circuit (page 2, lines 16-23, page 5, lines 17-18, page 5, line 39 to page 6, line 1). This universal bus provides a substantially fixed delay so that, as noted above in the comments for claim 1, the sync (or "control") pulse, LdTsExt, will propagate between circuits with a substantially fixed delay corresponding to the delay represented by the respective values loaded into comparator 302 (the target value) and the load register 304 (the future master timestamp value).

Claim 10 further calls for a slave counter to be synchronized at a future synchronization pulse generated independently of operations by the master counter and slave counter. In Dworkin, the slave counter is synchronized at a future synchronization pulse dependent on the operation of either the master counter or slave counter. In other words, at least one of the counters being synchronized is also one of the counters being called upon to generate the synchronization pulse triggering the synchronization event. Such self-referent operation can cause timing difficulties, as discussed above in connection with claim 1. In Dworkin '893, the same counter 315 that is being synchronized is also used to generate the internal pulse (Intpulse) that activates the synchronization event, that is, when the value of counter 315 matches the target

value stored in the compare register 305, the internal pulse activating synchronization is produced. In Dworkin '167, the same master counter 202 that is being synchronized with each slave circuit is also used to generate the synchronization (or "control") pulse (LdTsExt) used to activate the synchronization event, that is, when the value of master counter 202 matches the target value stored in the compare register 302, the synchronization (or "control") pulse is produced.

Based on these numerous differences, then, amended claim 10 patentably defines over the art of record.

Dependent claims 11-15

Dependent claims 11-15 incorporate each limitation of independent claim 10 and so patentably define over the art of record for the same reasons given above. However, dependent claim 11, even independently of its base claim 10, patentably defines over the art of record.

Dependent claim 11 specifies conditions under which a slave synchronization circuit will take over the operations of the master synchronization circuit. In particular, this is caused when an error message being sent to the slave circuit indicates that "the difference between the actual master timestamp value and the future master timestamp value," both corresponding to the future pulse in time, "is not within a predetermined range."

The specific paragraphs of Dworkin '893 that have been cited in the Action as relevant to claim 11's subject matter (i.e., pars. 0041, and 0086-0090) either discuss how the master and slave devices are interconnected by a bus (par. 0041), how latency is an indicator of elapsed time and important to synchronization requirements (par. 0086), or how each sync circuit is calibrated and likely starts at a different base value (par. 0087; this would include the original master circuit and each slave circuit), but how synchronization is achieved between circuits when each counter reaches its customized target value (FTSV; pars. 0084 and 0089) and is thereby triggered to produce an internal sync pulse and to load or reset its counter to the same future master timestamp value (TGCCCount(new); pars. 0088 and 0090-0091). There is no mention, in these cited paragraphs, of an "error" occurring in the master synchronization circuit. Dworkin '893 does say (in par. 0091, third-to-last sentence) that "one or more" of the slave devices 104B and 104C "can assume the load" of the master device 104A, when this master device "fails," but this can reasonably be interpreted, in the absence of further guidance, to mean catastrophic error, as indicated, for example, by complete signal loss or a burning smell. At best Dworkin teaches

identifying differences between the initial (calibration) counts of the master and slave devices (as presented at the output of the verify register 320 of each device) either as part of finding the future timestamp value, TGCCCount(new), for each counter (par. 0085, second to last paragraph) or, “periodically” for determining whether the devices remain in sync or whether, instead, synchronization needs to be repeated (par. 0093). Nothing in Dworkin ‘893 (or in ‘167, for that matter) shows or suggests, as claimed, signaling a slave circuit to take over operations as master with an error message... when the difference between the actual master timestamp value and the future master timestamp value (both corresponding to the future synchronization pulse) is not within a predetermined range. Accordingly, dependent claim 11, even independently of its base claim 10, patentably distinguishes over the art of record.

Independent claim 16 and dependent claims 17-20

Independent claim 16 is directed to a method for synchronizing circuitry. In particular, this method includes receiving an extrapolated master timestamp value for an upcoming time reference in an Internet Protocol (IP) packet over an asynchronous Internet connection. The method further includes comparing a generated local timestamp value at the upcoming time reference with the extrapolated master timestamp value and synchronizing ... according to the comparison.

Dworkin ‘893, in paragraph 0043 cited in the Action, does mention that cable modems 106 and 108 convert downstream signals (from the headend 102) into “IP data packets for receipt by an attached user device,” but these user devices, unlike the sync circuits 223 (par. 0048) that are provided in the headend MAC 218 (see Fig. 2 and circuit detail in Fig. 3), are not designed to perform a “synchronizing” operation. These sync circuits 223, on the other hand, while receiving an extrapolated master timestamp value (TGCCCount(new); pars. 0085 and 0088), receive this extrapolated value presumably over bus 112 designed to handle data exchanges between sync circuits (par. 0041, second-to-last sentence), but in any event over a fixed-delay path given that the extrapolated master timestamp value (TGCCCount(new)) is calculated based on a fixed “latency” between the master and slave circuits (par. 0085, second-to-last sentence). Using an “asynchronous” Internet connection to transfer the extrapolated master timestamp value to each sync circuit 223 would render the Dworkin system inoperable for its intended purpose. (In Dworkin ‘167, the universal bus 30 used to transfer the various timestamp values, each based on

TSLoadVal, and the master-generated sync pulse, LdTsExt, must likewise be a fixed-delay bus, for the reasons given above for claims 1 and 10.)

For that matter, in Dworkin '893 (and '167), the local timestamp value of each counter is not compared with the extrapolated master timestamp value and synchronized “according to the comparison;” rather, this local timestamp value is always reset regardless of what a comparison would show in terms of the local timestamp value matching the extrapolated master timestamp value. In Dworkin '893, the local (slave) timestamp value is compared, not to the “extrapolated master timestamp value” ($TCGCount(new) = FTSV\ of\ master = TCGCount(base)\ of\ master + Z$) that is loaded into load register 310, but rather to a target value customized for the local device ($= FTSV\ of\ the\ slave = TCGCount(base)\ of\ slave + Z$) that is loaded into compare register 305 (this is done so that each circuit will synchronize in “Z” cycles from its present base value; see last two sentences of par. 0083). The local timestamp value will always (eventually) advance to this target value (par. 0090, second and third sentences), so that the internal sync pulse (Intpulse in Fig. 3) will always be generated and counter 315, which provides the local timestamp value, will always be reset to the “extrapolated master timestamp value” held in load register 310 (regardless of whether the local timestamp value was already at this extrapolated master timestamp value).

Based on the above differences, independent claim 16 patentably defines over the art of record. Dependent claims 17-20, sharing every limitation of claim 16, likewise patentably define over the art of record.

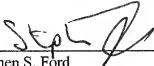
Conclusion

For the foregoing reasons, reconsideration and allowance of claims 1-22 of the application as amended is requested. The Examiner is encouraged to telephone the undersigned at (503) 224-2170 if it appears that an interview would be helpful in advancing the case.

Customer No. 73552

Respectfully submitted,

STOLOWITZ FORD COWGER LLP



Stephen S. Ford
Reg. No. 35,139

STOLOWITZ FORD COWGER LLP
621 SW Morrison Street, Suite 600
Portland, OR 97205
(503) 224-2170